AMENDMENTS TO THE CLAIMS

The following listing of claims replaces all prior versions and listings of claims in the application.

Listing of Claims:

8. (Currently Amended) A high-performance specification resolution method for use in detecting attacks against computer systems, comprising:

a) ~~formulating~~ receiving audit conditions to be detected using non-limiting specification formulas expressing fraudulent entry or attack patterns or abnormal operations, to be verified by examining ~~the~~ records of a log file of ~~the~~ a computer system;

b) expanding said formulas into subformulas for each record;

c) scanning ~~by an interpreter,~~ and generating, for each expanded formula ~~in each record~~, Horn clauses to resolve in order to detect whether or not the formula is valid ~~in the~~ for each record, the Horn clauses expressing ~~the~~ implications resolvent of the subformulas for each record scanned[[,]] in positive clauses~~, i.e. counting only~~ having a positive literal and in ~~non-positive~~ negative clauses having ~~, i.e. counting~~ at least one negative literal~~, which negative literals form the negative part of the clause~~;

d) storing the positive ~~Horn~~ clauses in a stack of ~~worked~~ subformulas, ~~and~~

storing, in a table of clauses, ~~comprising~~ a representation of~~, implicating subformula(s) constituting~~ the negative ~~part of the~~ clauses and ~~the link with the implicated subformula(s) constituting~~ the positive ~~part of the~~ clauses~~;, and~~

storing, in a table of counters, ~~the~~ a number of ~~formulas or subformulas present in the~~ negative literals in each negative ~~part of the~~ clause ~~for each implicated subformula~~;

e) resolving the table of clauses based on each positive clause ~~encountered,~~ so as to generate either an output file or an action of the computer system;

f) iterating steps b) through e) until the scanning of all the records in the log file is complete.

9. (Currently Amended) ~~A~~ The method according to claim 8, ~~characterized in that a~~ wherein temporal logic is used for the formulation of the specification.

10. (Currently Amended) ~~A~~ The method according to claim 8, ~~characterized in that~~ wherein the table of clauses is a matrix ~~and is~~ indexed in columns by subscripts of the formulas appearing in the negative ~~part of the Horn~~ clauses, and ~~the lines are~~ indexed in lines by subscripts of the formulas appearing in the positive ~~Horn~~ clauses ~~exactly~~.

11. (Currently Amended) ~~A~~ The method according to claim 8, ~~characterized in that~~ wherein the table of clauses is ~~preferably represented in the form of~~ a sparse matrix, the columns being represented by ~~means of~~ chained lists ~~and the implicit lines~~.

12. (Currently Amended) ~~A~~ The method according to claim 8, ~~characterized in that a step for~~ further comprising optimizing the expansion of the formulas ~~is obtained through~~ using a hash table to ensure that ~~the same~~ a formula is not expanded more than once in each record.

13. (Currently Amended) ~~A~~ The method according to claim 9, ~~characterized in that a step for~~ further comprising optimizing the expansion of the formulas ~~is obtained through~~ using a hash table to ensure that ~~the same~~ a formula is not expanded more than once in each record.

14. (Currently Amended) ~~A~~ The method according to claim 8, ~~characterized in that~~ wherein the log file is scanned only once from beginning to end.

15. (Currently Amended) A computer system comprising:

storage means; and

~~means~~ a processor, coupled to the storage means, for executing programs ~~for~~ implementing a high performance resolution method for ~~deleting~~ detecting attacks against the system wherein the ~~method~~ processor operates to:

a) ~~formulates~~ receive audit conditions to be detected using non-limiting specification formulas expressing fraudulent entry or attack patterns or abnormal operations, to be verified by examining the records of a log file ~~of the computer system~~;

b) expand[[s]] said formulas into subformulas for each record;

c) scan[[s]] ~~by an interpreter,~~ and generate[[s]], for each expanded formula ~~in each record~~, Horn clauses to resolve in order to detect whether or not the formula is valid ~~in the~~ for each record, the Horn clauses expressing ~~the~~ implications resolvent

of the subformulas for each record scanned[[,]] in positive clauses~~, i.e. counting only~~ having a positive literal, and in ~~non-positive~~ negative clauses~~, i.e. counting~~ having at least one negative literal~~, which negative literals form the negative part of the clause~~;

d) store[[s]] the positive ~~Horn~~ clauses in a stack of ~~worked~~ subformulas, ~~and storing,~~

store, in a table of clauses, ~~comprising~~ a representation of~~, implicating subformula(s) constituting~~ the negative ~~part of the~~ clauses and ~~the link with the implicated subformula(s) constituting~~ the positive ~~part of the~~ clauses, ~~and~~

store[[s]], in a table of counters, ~~the~~ a number of ~~formulas or subformulas present in the~~ negative literals in each negative ~~part of the~~ clause ~~for each implicated subformula~~; and

e) resolve[[s]] the table of clauses based on each positive clause ~~encountered~~, so as to generate either an output file or an action of the computer system~~;~~

~~an adaptor for translating information from a log file formulated in the specific language of the machine into a language comprehensible to an interpreter;~~

~~the interpreter receiving the information from the adapter and receiving the formulation of the specification in a temporal logic in a specification formula in order to expand said formula and fill in the table and the stack of worked subformulas stored in a memory of the computer system and resulting from the scanning of the computer system's log file;~~

~~a clause processing algorithm executed by the computer system, for resolving the Horn clauses using the information from the table and the stack of worked subformulas, said clause processing algorithm generating an output file or generating an action~~.

16. (Currently Amended) ~~A~~ The computer system ~~as defined in~~ according to claim 15 wherein ~~the~~ temporal logic is used for formulation of the specification.

17. (Currently Amended) ~~A~~ The computer system ~~as defined in~~ according to claim 15, wherein the table of clauses is a matrix ~~and is~~ indexed in columns by subscripts of the formulas appearing in the negative ~~part of the Horn~~ clauses, and ~~the lines are~~ in lines by subscripts of the formulas appearing in the positive ~~Horn~~ clauses ~~exactly~~.

18. (Currently Amended) ~~A~~ The computer system ~~as defined in~~ according to claim 15, wherein the table is ~~preferably represented in the form of~~ a sparse matrix, the columns being represented by ~~means of~~ chained lists ~~and the implicit lines~~.

19. (Currently Amended) ~~A~~ The computer system ~~as defined in~~ according to claim 15 including a hash table to ensure that ~~the same~~ a formula is not expanded more than once in each record.

20. (Currently Amended) ~~A~~ The computer system ~~as defined in~~ according to claim 16 including a hash table to ensure that ~~the same~~ a formula is not expanded more than once in each record.

21. (Currently Amended) ~~A~~ The computer system ~~as defined in~~ according to claim 15 including means for scanning the log file only once from beginning to end.

22. (New) The computer system according to claim 15, wherein the programs executed by the processor include:

an adaptor module for translating information from the log file;

an interpreter module for receiving the information from the adapter, receiving the specification formulas, expanding the specification formulas, and filling in the table of clauses, table of counters and the stack of ~~worked~~ subformulas stored in the storage means; and

a clause processing module for resolving the Horn clauses using the information from the table of clauses, the table of counters and the stack of worked subformulas, the clause processor generating the output file or generating the action.